6 Probabilistische Berechnungen

Eine *probabilistische Turingmaschine* (PTM) ist genau so definiert wie eine NTM. Es wird jedoch ein anderes Akzeptanzkriterium benutzt. Seien K, K' zwei Konfigurationen von M. Dann bezeichnet

$$Pr[K \to_M K'] = \begin{cases} \|\{K'' | K \to_M K''\}\|^{-1}, & K \to_M K' \\ 0, & \text{sonst} \end{cases}$$

die Wahrscheinlichkeit, dass M die Konfiguration K in einem Schritt in die Konfiguration K' überführt. Für eine Rechnung $\alpha = (K_1, K_2, \ldots, K_m)$ definieren wir

$$Pr[\alpha] \ = \ Pr[K_1 \to_M \cdots \to_M K_m] \quad = \quad \prod_{i=1}^{m-1} Pr[K_i \to_M K_{i+1}]$$

als die Wahrscheinlichkeit, dass M die Rechnung α ausführt. Weiter sei

$$Pr[M(x) \text{ akzeptiert}] = \sum_{\alpha} Pr[\alpha],$$

wobei sich die Summation über alle akzeptierenden Rechnungen α von M bei Eingabe x erstreckt. Die von einer PTM M akzeptierte Sprache ist

$$L(M) = \{x \in \Sigma^* \mid Pr[M(x) \text{ akzeptiert}] \geq 1/2\}.$$

Definition 79 (PP)
Eine Sprache $L \subseteq \Sigma^*$ gehört zur Klasse PP (probabilistic polynomial time), falls eine polynomiell zeitbeschränkte PTM (PPTM) M mit $L(M) = L$ existiert.

Satz 80
NP $\subseteq$ PP

Beweis: Sei $L \in$ NP und sei N eine polynomiell zeitbeschränkte NTM (NPTM) mit $L(N) = L$. Fassen wir N als PPTM auf, so gilt für alle $x \in \Sigma^*$,

$$\begin{aligned} x \in L \quad &\Rightarrow \quad Pr[N(x) \text{ akzeptiert}] \geq c^{-p(|x|)}, \\ x \notin L \quad &\Rightarrow \quad Pr[N(x) \text{ verwirft}] = 1, \end{aligned}$$

wobei c der maximale Verzweigungsgrad und p eine polynmielle Zeitschranke für N ist. Betrachte folgende PPTM N', die bei Eingabe x zufällig eine der beiden folgenden Möglichkeiten wählt:

- N' simuliert N bei Eingabe x,

- N' führt eine probabilistische Rechnung aus, bei der sie mit Wahrscheinlichkeit $1 - c^{-p(|x|)}$ akzeptiert (z.B. indem sie einen Zufallsstring $s = s_1 \cdots s_{p(|x|)} \in \{1, \ldots, c\}^{p(|x|)}$ auf's Band schreibt und nur im Fall $s = 1^{p(|x|)}$ verwirft).

Dann gilt für alle $x \in \Sigma^*$,

$$\mathit{Pr}[N'(x) \text{ akzeptiert}] = \frac{\mathit{Pr}[N(x) \text{ akzeptiert}] + 1 - c^{-p(|x|)}}{2}$$

und somit

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \mathit{Pr}[N'(x) \text{ akzeptiert}] \geq \frac{c^{-p(|x|)} + 1 - c^{-p(|x|)}}{2} = 1/2, \\ x \notin L \quad &\Rightarrow \quad \mathit{Pr}[N'(x) \text{ akzeptiert}] = \frac{1 - c^{-p(|x|)}}{2} < 1/2. \end{aligned}$$

■

Es ist leicht zu sehen, dass folgendes Problem PP-vollständig ist.

MajoritySat (MajSat):

Gegeben: Eine boolsche Formel $F(x_1, \ldots, x_n)$.

Gefragt: Wird F von mindestens der Hälfte aller 2^n Belegungen erfüllt?

Lemma 81

Für jede Sprache $L \in$ PP existiert eine PPTM M, die bei keiner Eingabe $x \in \Sigma^*$ mit Wahrscheinlichkeit $1/2$ akzeptiert.

Beweis: Nach Definition von PP existieren eine PPTM N, ein Polynom p und eine Konstante $c \geq 1$ mit

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \mathit{Pr}[N(x) \text{ akzeptiert}] \geq 1/2, \\ x \notin L \quad &\Rightarrow \quad \mathit{Pr}[N(x) \text{ akzeptiert}] \leq 1/2 - c^{-p(|x|)}. \end{aligned}$$

Sei N' eine PPTM mit $\mathit{Pr}[N'(x) \text{ akzeptiert}] = 1/2(1 + \epsilon)$, wobei $\epsilon = c^{-p(|x|)}$, und betrachte folgende PPTM M, die bei Eingabe x zufällig eine der beiden folgenden Möglichkeiten wählt:

- M simuliert N bei Eingabe x,
- M simuliert N' bei Eingabe x.

Dann gilt

$$\mathit{Pr}[M(x) \text{ akzeptiert}] = \frac{\mathit{Pr}[N(x) \text{ akzeptiert}] + \mathit{Pr}[N'(x) \text{ akzeptiert}]}{2}$$

also

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ akzeptiert}] \geq \frac{1/2 + 1/2(1 + \varepsilon)}{2} = 1/2 + \varepsilon/4 > 1/2 \\ x \notin L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ akzeptiert}] \leq \frac{1/2 - \varepsilon + 1/2(1 + \varepsilon)}{2} = 1/2 - \varepsilon/4 < 1/2. \end{aligned}$$

■

Nun können wir den Abschluss von PP unter symmetrischer Differenz (und damit unter Komplement) zeigen.

Satz 82
PP ist unter symmetrischer Differenz abgeschlossen,

$$L_1, L_2 \in \text{PP} \Rightarrow L_1 \triangle L_2 = (L_1 \setminus L_2) \cup (L_2 \setminus L_1) \in \text{PP}.$$

Beweis: Nach obigem Lemma existieren PPTMs M_1 und M_2 mit

$$\begin{array}{lcl} x \in L_i & \Rightarrow & \mathit{Pr}[M_i(x) \text{ akzeptiert}] = 1/2 + \epsilon_i, \\ x \notin L_i & \Rightarrow & \mathit{Pr}[M_i(x) \text{ akzeptiert}] = 1/2 - \epsilon_i. \end{array}$$

wobei $\epsilon_i > 0$ ist und von x abhängen darf. Betrachte folgende PPTM M:

> M simuliert bei Eingabe x zunächst $M_1(x)$ und dann $M_2(x)$ und akzeptiert, falls genau eine der beiden Maschinen akzeptiert.

Dann gilt:

$$\begin{array}{rcl} \mathit{Pr}[M(x) \text{ akzeptiert}] & = & \mathit{Pr}[M_1(x) \text{ akzeptiert}] \cdot \mathit{Pr}[M_2(x) \text{ verwirft}], \\ & + & \mathit{Pr}[M_1(x) \text{ verwirft}] \cdot \mathit{Pr}[M_2(x) \text{ akzeptiert}]. \end{array}$$

Dann ist die Akzeptanzwahrscheinlichkeit von $M(x)$ im Fall $x \in L_1 \setminus L_2$,

$$\begin{array}{rcl} \mathit{Pr}[M(x) \text{ akzeptiert}] & = & (1/2 + \epsilon_1)(1/2 + \epsilon_2) + (1/2 - \epsilon_1)(1/2 - \epsilon_2) \\ & = & (1/2 + 2\epsilon_1\epsilon_2) > 1/2 \end{array}$$

und analog im Fall $x \in L_2 \setminus L_1$,

$$\mathit{Pr}[M(x) \text{ akzeptiert}] = 1/2 + 2\epsilon_1\epsilon_2 > 1/2.$$

Dagegen ergibt sich im Fall $x \in L_1 \cap L_2$,

$$\begin{array}{rcl} \mathit{Pr}[M(x) \text{ akzeptiert}] & = & (1/2 + \epsilon_1)(1/2 - \epsilon_2) + (1/2 - \epsilon_1)(1/2 + \epsilon_2) \\ & = & (1/2 - 2\epsilon_1\epsilon_2) < 1/2 \end{array}$$

und analog im Fall $x \notin L_1 \cup L_2$,

$$\mathit{Pr}[M(x) \text{ akzeptiert}] = 1/2 - 2\epsilon_1\epsilon_2 < 1/2.$$

■

Korollar 83
PP = co-PP.

Anfang der 90er Jahre konnte auch der Abschluss von PP unter Schnitt und Vereinigung bewiesen werden.

Definition 84 (BPP, RP und ZPP)

- Eine Sprache L gehört zu BPP (bounded error probalistic polynomial time), falls eine PPTM M existiert mit

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ akzeptiert}] \geq \frac{2}{3}, \\ x \notin L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ verwirft}] \geq \frac{2}{3}. \end{aligned}$$

- L gehört zu RP (random polynomial time), falls eine PPTM M existiert mit

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ akzeptiert}] \geq 1/2, \\ x \notin L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ verwirft}] = 1. \end{aligned}$$

- L gehört zu ZPP (zero error probalistic polynomial time), falls eine PPTM M existiert mit

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ verwirft}] = 0 \wedge \mathit{Pr}[M(x) \text{ akzeptiert}] \geq 1/2, \\ x \notin L \quad &\Rightarrow \quad \mathit{Pr}[M(x) \text{ akzeptiert}] = 0 \wedge \mathit{Pr}[M(x) \text{ verwirft}] \geq 1/2. \end{aligned}$$

Man beachte, dass wir im Falle von RP und BPP o.B.d.A. davon ausgehen können, dass M am Ende jeder Rechnung entweder akzeptiert (Endzustand ist q_{ja}) oder verwirft (Endzustand ist q_{nein}). Eine solche PPTM M wird auch als eine Maschine vom Typ „**Monte Carlo**“ bezeichnet, da M ein falsches Ergebnis liefern kann. Allerdings ist die Wahrscheinlichkeit, dass M lügt (also im Fall $x \in L$ nicht akzeptiert bzw. im Fall $x \notin L$ nicht verwerft), beschränkt:

$$\forall x : \mathit{Pr}[M(x) \neq \chi_L(x)] \leq 1/3.$$

Im Unterschied zu einer BPP-Maschine, die sowohl im Fall $x \in L$ als auch im Fall $x \notin L$ lügen darf (also so genannte **zweiseitige** Fehler machen kann), sind einer RP-Maschine M nur (so genannte **einseitige**) Fehler im Fall $x \in L$ erlaubt:

$$\begin{aligned} &\forall x \in L : \mathit{Pr}[M(x) \neq \chi_L(x)] \leq 1/2, \\ &\forall x \notin L : \mathit{Pr}[M(x) \neq \chi_L(x)] = 0. \end{aligned}$$

Dagegen darf eine ZPP-Maschine M überhaupt nicht „lügen“. Sie darf höchstens im Zustand q_h halten, womit jedoch keine Behauptung über den Status von x (d.h. ob x zur Sprache gehört oder nicht) verbunden ist:

$$\forall x : \mathit{Pr}[M(x) \neq \chi_L(x)] = 0 \text{ und } \mathit{Pr}[M(x) \text{ hält im Zustand } q_h] \leq 1/2.$$

Solche PPTMs werden auch als Maschinen vom Typ „**Las Vegas**“ bezeichnet.

Satz 85

ZPP $=$ RP $\cap$ co-RP.

Beweis: Die Inklusion von links nach rechts folgt direkt aus den Definitionen. Für die umgekehrte Richtung sei $L \in \mathrm{RP} \cap \text{co-RP}$ und seien M_1 und M_2 RP-Maschinen für L und $\overline{L}$, wobei wir annehmen, dass M_1 und M_2 niemals im Zustand q_h halten. Weiter sei $\overline{M}_2$ die PPTM, die durch Vertauschen der beiden Endzustände q_{ja} und q_{nein} aus M_2 hervorgeht (man könnte $\overline{M}_2$ als eine co-RP-Maschine für L bezeichnen). Dann gilt

$$\begin{aligned} x \in L &\quad\Rightarrow\quad \mathit{Pr}[M_1(x) \text{ akzeptiert}] \geq 1/2 \wedge \mathit{Pr}[\overline{M}_2(x) \text{ akzeptiert}] = 1, \\ x \notin L &\quad\Rightarrow\quad \mathit{Pr}[M_1(x) \text{ verwirft}] = 1 \wedge \mathit{Pr}[\overline{M}_2(x) \text{ verwirft}] \geq 1/2. \end{aligned}$$

Hieraus ergeben sich folgende Implikationen:

$$\begin{aligned} M_1(x) \text{ akzeptiert} &\quad\Rightarrow\quad x \in L, \\ \overline{M}_2(x) \text{ verwirft} &\quad\Rightarrow\quad x \notin L. \end{aligned}$$

Betrachte folgende PPTM M:

> M simuliert bei Eingabe x die beiden PPTMs $M_1(x)$ und $\overline{M}_2(x)$ und hält in einem Zustand gemäß folgender Tabelle (man beachte, dass die Kombination „$M_1(x)$ akzeptiert“ und „$\overline{M}_2(x)$ verwirft“ nicht auftreten kann):

		$\overline{M}_2$	
		akzeptiert	verwirft
M_1	akzeptiert	q_{ja}	–
	verwirft	q_h	q_{nein}

Dann gilt:

$$\begin{aligned} M(x) \text{ akzeptiert} &\quad\Leftrightarrow\quad M_1(x) \text{ akzeptiert}, \\ M(x) \text{ verwirft} &\quad\Leftrightarrow\quad \overline{M}_2(x) \text{ verwirft} \end{aligned}$$

und daher

$$\begin{aligned} x \in L \quad\Rightarrow\quad & \mathit{Pr}[M(x) \text{ akzeptiert}] = \mathit{Pr}[M_1(x) \text{ akzeptiert}] \geq 1/2 \\ & \mathit{Pr}[M(x) \text{ verwirft}] = \mathit{Pr}[\overline{M}_2(x) \text{ verwirft}] = 0 \\ x \notin L \quad\Rightarrow\quad & \mathit{Pr}[M(x) \text{ verwirft}] = \mathit{Pr}[\overline{M}_2(x) \text{ verwirft}] \geq 1/2 \\ & \mathit{Pr}[M(x) \text{ akzeptiert}] = \mathit{Pr}[M_1(x) \text{ akzeptiert}] = 0 \end{aligned}$$

Dies zeigt, dass M eine ZPP-Maschine für L ist. ∎

6.1 Reduktion der Fehlerwahrscheinlichkeit

In diesem Abschnitt zeigen wir, wie sich für RP-, ZPP- und BPP-Maschinen M die Fehlerwahrscheinlichkeit $\mathit{Pr}[M(x) \neq \chi_L(x)]$ auf einen exponentiell kleinen Wert $2^{-q(|x|)}$ reduzieren lässt. Wir betrachten zunächst den Fall einer RP-Maschine.

Satz 86

Sei q ein beliebiges Polynom. Dann existiert zu jeder Sprache $L \in$ RP eine PPTM M mit

$$\begin{aligned} x \in L &\Rightarrow \mathit{Pr}[M(x) \text{ akzeptiert}] \geq 1 - 2^{-q(|x|)}, \\ x \notin L &\Rightarrow \mathit{Pr}[M(x) \text{ verwirft}] = 1. \end{aligned}$$

Beweis: Sei M eine RP-Maschine für L, d.h. M ist eine PPTM mit

$$\begin{aligned} x \in L &\Rightarrow \mathit{Pr}[M(x) \text{ verwirft}] \leq 1/2, \\ x \notin L &\Rightarrow \mathit{Pr}[M(x) \text{ verwirft}] = 1. \end{aligned}$$

Betrachte die PPTM M', die $q(|x|)$ Simulationen von M ausführt und nur dann ihre Eingabe x verwirft, wenn M sie bei allen $q(|x|)$ Simulationen verwirft, und andernfalls akzeptiert. Dann gilt

$$\begin{aligned} x \in L &\Rightarrow \mathit{Pr}[M(x) \text{ verwirft}] \leq 1/2 \Rightarrow \mathit{Pr}[M'(x) \text{ verwirft}] \leq 2^{-q(|x|)}, \\ x \notin L &\Rightarrow \mathit{Pr}[M(x) \text{ verwirft}] = 1 \Rightarrow \mathit{Pr}[M'(x) \text{ verwirft}] = 1. \end{aligned}$$

■

Ganz analog lässt sich die Zuverlässigkeit einer ZPP-Maschine verbessern.

Satz 87

Sei q ein beliebiges Polynom. Dann existiert zu jeder Sprache $L \in$ ZPP eine PPTM M mit

$$\forall x : \mathit{Pr}[M(x) \neq \chi_L(x)] = 0 \text{ und } \mathit{Pr}[M(x) \text{ hält im Zustand } q_h] \leq 2^{-q(|x|)}.$$

Für die Reduktion der Fehlerwahrscheinlichkeit von BPP-Maschinen benötigen wir das folgende Lemma.

Lemma 88

Sei E ein Ereignis, das mit Wahrscheinlichkeit $1/2 - \epsilon$, $\epsilon > 0$, auftritt. Dann ist die Wahrscheinlichkeit, dass sich E bei $m = 2t + 1$ unabhängigen Wiederholungen mindestens $(t + 1)$-mal ereignet, höchstens $1/2(1 - 4\epsilon^2)^t$.

Beweis: Für $i = 1, \ldots, m$ sei X_i die Indikatorvariable

$$X_i = \begin{cases} 1, & \text{Ereignis } E \text{ tritt beim } i\text{-ten Versuch ein,} \\ 0, & \text{sonst} \end{cases}$$

und X sei die Zufallsvariable $X = \sum_{i=1}^{m} X_i$. Dann ist X binomial verteilt mit Parametern m und $p = 1/2 - \epsilon$. Folglich gilt für $i > m/2$,

$$\begin{aligned} \mathit{Pr}[X = i] &= \binom{m}{i} (1/2 - \epsilon)^i (1/2 + \epsilon)^{m-i} \\ &= \binom{m}{i} (1/2 - \epsilon)^{m/2} (1/2 + \epsilon)^{m/2} \left(\frac{1/2 - \epsilon}{1/2 + \epsilon}\right)^{i-m/2} \\ &\leq \binom{m}{i} \underbrace{(1/2 - \epsilon)^{m/2} (1/2 + \epsilon)^{m/2}}_{(1/4 - \epsilon^2)^{m/2}} \end{aligned}$$

Wegen

$$\sum_{i=t+1}^{m} \binom{m}{i} \leq 2^{m-1} = \frac{4^{m/2}}{2}$$

erhalten wir somit

$$\sum_{i=t+1}^{m} Pr[X = i] \quad \leq \quad \left(1/4 - \epsilon^2\right)^{\frac{m}{2}} \sum_{i=t+1}^{m} \binom{m}{i} \quad \leq \quad \frac{(1 - 4\epsilon^2)^{\frac{m}{2}}}{2} \quad \leq \quad \frac{(1 - 4\epsilon^2)^t}{2}.$$

■

Satz 89

Sei q ein beliebiges Polynom. Dann existiert zu jeder Sprache $L \in$ BPP eine PPTM M mit

$$Pr[M(x) \neq \chi_L(x)] \leq 2^{-q(|x|)}.$$

Beweis: Sei M eine BPP-Maschine für L, d.h.

$$Pr[\underbrace{M(x) \neq \chi_L(x)}_{E}] \leq 1/3 = 1/2 - 1/6.$$

Betrachte die PPTM M', die bei Eingabe x, $|x| = n$, $m = 2t(n) + 1$ Simulationen von $M(x)$ ausführt, wobei $t(n) = (q(n) - 1)/\log_2(9/8)$ ist, und x akzeptiert, falls M bei mindestens $t(n) + 1$ dieser Simulationen akzeptiert. Dann folgt nach obigem Lemma

$$\begin{aligned} Pr[M(x) \neq \chi_L(x)] \quad &= \quad Pr\,[\text{Ereignis } E \text{ tritt mindestens } (t(n) + 1)\text{-mal ein}] \\ &\leq \quad 1/2\,(1 - 4/36)^{t(n)} \\ &\leq \quad 1/2\,(8/9)^{t(n)} \\ &= \quad 2^{-q(|x|)}. \end{aligned}$$

■

Satz 90

BPP $\subseteq$ PSK.

Beweis: Sei $L \in$ BPP und sei M eine BPP-Maschine für L. Nach vorigem Satz können wir annehmen, dass

$$Pr[M(x) \neq \chi_L(x)] < 2^{-n}$$

ist. Weiter sei p eine polynomiale Zeitschranke und $c \in \mathbb{N}$ der maximale Verzweigungsgrad von M. Setzen wir $k := \mathrm{kgV}(2, 3, \ldots, c)$, dann können wir für eine gegebene Eingabelänge n jeder Folge $r = r_1 \cdots r_{p(n)}$ aus der Menge $R_n = \{1, \ldots, k\}^{p(n)}$ eindeutig eine Rechnung von $M(x)$ zuordnen, indem wir im i-ten Rechenschritt aus den c_i zur Auswahl stehenden Folgekonfigurationen die $(r_i \bmod c_i)$-te wählen. Bezeichnen wir das Ergebnis der so beschriebenen Rechnung mit $M_r(x)$, so gilt

$$Pr[M(x) \neq \chi_L(x)] = Pr_{r \in_R R_n}[M_r(x) \neq \chi_L(x)] < 2^{-n}.$$

Daher folgt

$$\mathit{Pr}[\exists x \in \{0,1\}^n : M(x) \neq \chi_L(x)] \quad \leq \sum_{x \in \{0,1\}^n} \mathit{Pr}_{r \in_R R_n}[M_r(x) \neq \chi_L(x)] \ < \ 1.$$

Also muss für jede Eingabelänge n eine Folge r_n existieren, so dass M_{r_n} alle Eingaben der Länge n korrekt entscheidet. Wie wir gesehen haben, kann die polynomiell zeitbeschränkte Rechnung von M_{r_n} durch einen Schaltkreis polynomieller Größe simuliert werden. ■

7 Die Polynomialzeithierachie

7.1 Anzahl-Operatoren

Definition 91 (Anzahlklassen)
Sei $\mathcal{C}$ eine Sprachklasse und sei p ein Polynom. Eine Funktion $f : \Sigma^* \to \mathbb{N}$ gehört zur Anzahlklasse $\#_p\mathcal{C}$, falls eine Sprache $B \in \mathcal{C}$ existiert mit

$$f(x) = \|\{y \in \{0,1\}^{p(|x|)} \mid x\#y \in B\}\|.$$

Der Funktionswert $f(x)$ bestimmt also die Anzahl der Paare $x\#y \in B$ mit $|y| = p(|x|)$. Ferner definieren wir folgende Sprachklassen (n bezeichnet die Länge von x):

$$\begin{aligned}
\exists_p\mathcal{C} &= \{A \mid \exists f \in \#_p\mathcal{C} : x \in A \Leftrightarrow f(x) > 0\}\\
\forall_p\mathcal{C} &= \{A \mid \exists f \in \#_p\mathcal{C} : x \in A \Leftrightarrow f(x) = 2^{p(n)}\}\\
\mathrm{R}_p\mathcal{C} &= \{A \mid \exists f \in \#_p\mathcal{C} : x \in A \Leftrightarrow f(x) \geq 1/2 \cdot 2^{p(n)} \wedge x \notin A \Leftrightarrow f(x) = 0\}\\
\mathrm{BP}_p\mathcal{C} &= \{A \mid \exists f \in \#_p\mathcal{C} : x \in A \Leftrightarrow f(x) \geq 2/3 \cdot 2^{p(n)} \wedge x \notin A \Leftrightarrow f(x) \leq 1/3 \cdot 2^{p(n)}\}\\
\mathrm{P}_p\mathcal{C} &= \{A \mid \exists f \in \#_p\mathcal{C} : x \in A \Leftrightarrow f(x) \geq 1/2 \cdot 2^{p(n)}\}\\
\oplus_p\mathcal{C} &= \{A \mid \exists f \in \#_p\mathcal{C} : x \in A \Leftrightarrow f(x) \text{ ist ungerade}\}
\end{aligned}$$

Für einen beliebigen Operator $\mathrm{Op} \in \{\#, \exists, \forall, \mathrm{R}, \mathrm{BP}, \mathrm{P}, \oplus\}$ bezeichnen wir mit $\mathrm{Op} \cdot \mathcal{C}$ die Vereinigung

$$\mathrm{Op} \cdot \mathcal{C} = \bigcup_p \mathrm{Op}_p\mathcal{C}$$

über alle Polynome p.

Proposition 92
(i) $\exists \cdot \mathrm{P} = \mathrm{NP} = \exists \cdot \mathrm{NP}$,

(ii) $\forall \cdot \mathrm{P} = \text{co-NP} = \forall \cdot \text{co-NP}$.

Bei der Definition von PTMs M haben wir zwar den maximalen Verzweigungsgrad c nicht beschränkt. Das folgende Lemma zeigt jedoch, dass wir o.B.d.A. $c = 2$ annehmen können.

Lemma 93
Für jede PPTM M existiert eine PPTM M' mit maximalem Verzweigungsgrad 2 und

$$Pr[M'(x) \text{ akzeptiert}] - 1/2 = \beta(x)(Pr[M(x) \text{ akzeptiert}] - 1/2),$$

wobei $\beta(x) \geq 1/2$ ist.

Beweis: Sei M eine PPTM, p eine polynomielle Zeitschranke und $c \in \mathbb{N}$ der maximale Verzweigungsgrad von M. Wie im Beweis der Inklusion von BPP in PSK können wir die Rechnungen von M bei Eingaben der Länge n durch Folgen $r \in \{1, \dots, k\}^{p(n)}$ beschreiben, so dass

$$\mathit{Pr}[M(x) \text{ akzeptiert}] = \mathit{Pr}_{r \in_R R_n}[M_r(x) \text{ akzeptiert}]$$

ist. Sei nun $l(n) = \lceil \log_2(k^{p(n)}) \rceil$ und sei $D_n \subseteq \{0,1\}^{l(n)}$ die Menge der ersten $k^{p(n)}$ Binärstrings der Länge $l(n)$. Dann können wir die Binärstrings in D_n zur Kodierung der Folgen $r \in R_n$ benutzen. Betrachte folgende PPTM M'.

> M' rät bei Eingabe x, $|x| = n$, zufällig einen Binärstring $d \in \{0,1\}^{l(n)}$. Gehört d zu D_n, so berechnet M' die zugehörige Folge $r \in R_n$ und verhält sich wie $M_r(x)$. Andernfalls akzeptiert M' mit Wahrscheinlichkeit $1/2$.

Da mehr als die Hälfte aller Strings der Länge $l(n)$ in D_n enthalten ist, ist $\beta(x) := \mathit{Pr}[d \in D_n] > 1/2$, und es folgt

$$\begin{aligned}
\mathit{Pr}[M'(x) \text{ akzeptiert}] &= \mathit{Pr}[d \in D_n] \cdot \underbrace{\mathit{Pr}[M'(x) \text{ akzeptiert} \mid d \in D_n]}_{=\mathit{Pr}[M(x) \text{ akzeptiert}]} \\
&\quad + \mathit{Pr}[d \notin D_n] \cdot \underbrace{\mathit{Pr}[M'(x) \text{ akzeptiert} \mid d \notin D_n]}_{=1/2} \\
&= \mathit{Pr}[d \in D_n](\mathit{Pr}[M(x) \text{ akzeptiert}] - 1/2) \\
&\quad + 1/2(\underbrace{\mathit{Pr}[d \in D_n] + \mathit{Pr}[d \notin D_n]}_{=1}) \\
&= \beta(x)(\mathit{Pr}[M(x) \text{ akzeptiert}] - 1/2) + 1/2.
\end{aligned}$$

■

Korollar 94

(i) R · P = RP = R · RP,

(ii) BP · P = BPP = BP · BPP,

(iii) P · P = PP.

Allgemeiner gilt für jede Sprachklasse $\mathcal{C}$,

$$\mathrm{R} \cdot \mathrm{R} \cdot \mathcal{C} = \mathrm{R} \cdot \mathcal{C},$$

falls $\mathcal{C}$ unter disjunktiven Reduktionen abgeschlossen ist, und

$$\mathrm{BP} \cdot \mathrm{BP} \cdot \mathcal{C} = \mathrm{BP} \cdot \mathcal{C},$$

falls $\mathcal{C}$ unter majority-Reduktionen abgeschlossen ist. Dabei ist A auf B **disjunktiv reduzierbar** (in Zeichen: $A \leq_{disj} B$), falls eine Funktion $f \in$ FP existiert, die für jedes Wort x eine Liste $y_1\# \cdots \#y_m$ von Wörtern y_i liefert mit

$$x \in A \Leftrightarrow \exists i \in \{1, \dots, m\} : y_i \in B$$

bzw.

$$x \in A \Leftrightarrow \|\{i \in \{1, \ldots, m\} \mid y_i \in B\} \geq {}^{m}\!/_{2}$$

im Fall einer **majority-Reduktion**, wofür wir kurz $A \leq_{maj} B$ schreiben. Es ist leicht zu sehen, dass die Klassen P, NP und co-NP unter beiden Typen von Reduktionen abgeschlossen sind. Folglich ist auch die Klasse BP $\cdot$ NP unter dem BP-Operator abgeschlossen.

Definition 95 (Polynomialzeithierachie)
Die Polynomialzeithierachie besteht aus den Stufen Σ_k^p und Π_k^p, $k \geq 0$, welche induktiv wie folgt definiert sind:

$$\begin{aligned} \Sigma_0^p &= \mathrm{P}, & \Pi_0^p &= \mathrm{P}, \\ \Sigma_{k+1}^p &= \exists \cdot \Pi_k^p, & \Pi_{k+1}^p &= \forall \cdot \Sigma_k^p, \quad k \geq 0. \end{aligned}$$

Die Vereinigung aller Stufen der Polynomialzeithierachie bezeichnen wir mit PH,

$$\mathrm{PH} = \bigcup_{k\geq 0} \Sigma_k^p = \bigcup_{k\geq 0} \Pi_k^p.$$

Es ist leicht zu sehen, dass $\Sigma_k^p = \text{co-}\Pi_k^p$ ist. Es ist nicht bekannt, ob die Polynomialzeithierachie echt ist, also $\Sigma_k^p \neq \Sigma_{k+1}^p$ für alle $k \geq 0$ gilt. Die Annahme $\Sigma_k^p = \Sigma_{k+1}^p$ ist mit einem Kollaps von PH auf die k-te Stufe äquivalent. Es gilt allerdings als unwahrscheinlich, dass die Polynomialzeithierachie auf eine kleine Stufe kollabiert.

Satz 96
Für alle $k \geq 0$ gilt: $\Sigma_k^p = \Sigma_{k+1}^p \Leftrightarrow \Sigma_k^p = \Pi_k^p \Leftrightarrow \mathrm{PH} = \Sigma_k^p$.

Beweis: Wegen $\Pi_k^p \subseteq \Sigma_{k+1}^p$ impliziert die Inklusion $\Sigma_k^p = \Sigma_{k+1}^p$ sofort $\Pi_k^p \subseteq \Sigma_k^p$, was mit $\Sigma_k^p = \Pi_k^p$ gleichbedeutend ist. Für die zweite Implikation zeigen wir durch Induktion über k, dass unter der Voraussetzung $\Sigma_k^p = \Pi_k^p$ alle Stufen Σ_l^p, $l \geq k$, in Σ_k^p enthalten sind. Der Induktionsanfang $l = k$ ist klar. Für den Induktionsschritt setzen wir die Gleichheit $\Sigma_l^p = \Sigma_k^p$ (bzw. $\Pi_l^p = \Pi_k^p$) voraus und folgern

$$\Sigma_{l+1}^p = \exists \cdot \Pi_l^p = \exists \cdot \Pi_k^p = \exists \cdot \Sigma_k^p = \Sigma_k^p.$$

Die Implikation $\mathrm{PH} = \Sigma_k^p \Rightarrow \Sigma_k^p = \Sigma_{k+1}^p$ ist klar. ■

Als Folgerung hieraus ergibt sich, dass eine NP-vollständige Sprache nicht in P (bzw. co-NP) enthalten ist, außer wenn PH auf P bzw. NP kollabiert. Als nächstes wollen wir zeigen, dass NP-vollständige Sprachen nicht in BP$\cdot$co-NP $=$ co-BP$\cdot$NP enthalten sind, außer wenn PH $=$ BP $\cdot$ NP ist. Hierfür benötigen wir die folgenden Abschlusseigenschaften der Klasse BP $\cdot$ NP.

Lemma 97
Sei $\mathcal{C}$ eine unter $\leq_{maj}$ abgeschlossene Sprachklasse. Dann gilt

$$\exists \cdot \mathrm{BP} \cdot \mathcal{C} \subseteq \mathrm{BP} \cdot \exists \cdot \mathcal{C}$$

Folglich ist BP $\cdot$ NP unter dem $\exists$-Operator abgeschlossen.

Beweis: Sei $L \in \exists \cdot \mathrm{BP} \cdot \mathcal{C}$, d.h. es existieren Polynome p und q sowie eine Sprache $B \in \mathcal{C}$ mit (wobei $n = |x|$ ist)

$$x \in L \Rightarrow \exists y \in \{0,1\}^{p(n)} : \|\{z \in \{0,1\}^{q(n)} \mid x\#y\#z \in B\}\| \geq (1 - 2^{-p(n)-2})2^{q(n)},$$
$$x \notin L \Rightarrow \forall y \in \{0,1\}^{p(n)} : \|\{z \in \{0,1\}^{q(n)} \mid x\#y\#z \in B\}\| \leq (2^{-p(n)-2})2^{q(n)}.$$

Dann folgt

$$\begin{aligned} x \in L \Rightarrow \|\{z \in \{0,1\}^{q(n)} \mid \exists y \in \{0,1\}^{p(n)} : x\#y\#z \in B\}\| &\geq (1 - 2^{-p(n)-2})2^{q(n)}, \\ &\geq ({}^2\!/\!_3)2^{q(n)}, \\ x \notin L \Rightarrow \|\{z \in \{0,1\}^{q(n)} \mid \exists y \in \{0,1\}^{p(n)} : x\#y\#z \in B\}\| &\leq 2^{p(n)}(2^{-p(n)-2})2^{q(n)} \\ &= ({}^1\!/\!_3)2^{q(n)}. \end{aligned}$$

■

Satz 98
NP $\subseteq$ BP $\cdot$ co-NP $\Rightarrow$ PH $=$ BP $\cdot$ NP.

Beweis: Gelte NP $\subseteq$ BP $\cdot$ co-NP. Wir zeigen durch Induktion über k, dass dann auch die Klassen Σ_k^p, $k \geq 0$, in BP $\cdot$ co-NP enthalten sind. Der Induktionsanfang $k = 0$ ist klar. Für den Induktionsschritt setzen wir die Inklusionen NP $\subseteq$ BP $\cdot$ co-NP und $\Sigma_k^p \subseteq$ BP $\cdot$ co-NP (was mit $\Pi_k^p \subseteq$ BP $\cdot$ NP gleichbedeutend ist) voraus und folgern

$$\begin{aligned} \Sigma_{k+1}^p &= \exists \cdot \Pi_k^p \\ &\subseteq \exists \cdot \mathrm{BP} \cdot \mathrm{NP} \\ &\subseteq \mathrm{BP} \cdot \exists \cdot \mathrm{NP} \\ &= \mathrm{BP} \cdot \mathrm{NP} \\ &\subseteq \mathrm{BP} \cdot \mathrm{BP} \cdot \text{co-NP} \\ &= \mathrm{BP} \cdot \text{co-NP}. \end{aligned}$$

■

Zum Abschluss dieses Kapitels zeigen wir, dass BPP in der zweiten Stufe der Polynomialzeithierarchie enthalten ist.

Satz 99
BPP $\subseteq \Sigma_2^p \cap \Pi_2^p$.

Beweis: Sei $A \in$ BPP. Dann existiert eine Sprache $B \in$ P und ein Polynom p mit

$$Pr_{y \in_R \{0,1\}^{p(n)}}[x \in L \Leftrightarrow x\#y \in B] \geq 1 - 2^{-n}.$$

Sei $\oplus$ die bitweise XOR-Operation auf $\{0,1\}^n$, d.h.

$$x_1 \cdots x_n \oplus y_1 \cdots y_n = z_1 \cdots z_n, \text{ wobei } z_i = x_i \oplus y_i.$$

Wie wir gleich sehen werden, können wir dann die Zugehörigkeit von x zu A durch

$$x \in A \Leftrightarrow \exists u_1 \cdots u_{p(n)} \in \{0,1\}^{p(n)} \forall v \in \{0,1\}^{p(n)} \exists i : x\#(v \oplus u_i) \in B \tag{7.1}$$

charakterisieren. Dies beweist, dass A zu Σ_2^p gehört, da die Sprache

$$B' = \{x\#u_1\# \cdots \#u_{p(n)}\#v \mid \exists i : x\#(v \oplus u_i) \in B\}$$

in P entscheidbar ist. Wir zeigen zuerst die Richtung von links nach rechts der Äquivalenz (7.1). Für $x \in A$ enthält die Menge

$$B(x) = \{y \in \{0,1\}^{p(n)} \mid x\#y \in B\}$$

mindestens $(1 - 2^{-n})2^{p(n)}$ Wörter. Daher gilt für alle $v \in \{0,1\}^{p(n)}$,

$$\mathit{Pr}_{u \in_R \{0,1\}^{p(n)}}[v \oplus u \notin B(x)] \leq 2^{-n}.$$

Folglich ist

$$\mathit{Pr}_{u_1,\ldots,u_{p(n)} \in_R \{0,1\}^{p(n)}}[\forall i : v \oplus u_i \notin B(x)] \leq (2^{-n})^{p(n)} = 2^{-np(n)}.$$

und somit

$$\mathit{Pr}_{u_1,\ldots,u_{p(n)} \in_R \{0,1\}^{p(n)}}[\exists v \in \{0,1\}^{p(n)} \forall i : v \oplus u_i \notin B(x)] \leq 2^{p(n)} 2^{-np(n)}.$$

Da diese Wahrscheinlichkeit für $n > 1$ kleiner als 1 ist, müssen für jedes x mit $|x| > 1$ also Wörter $u_1, \ldots, u_{p(n)} \in \{0,1\}^{p(n)}$ mit der gewünschten Eigenschaft existieren.

Zum Nachweis der Rückrichtung nehmen wir an, dass Wörter $u_1, \ldots, u_{p(n)} \in \{0,1\}^{p(n)}$ existieren, so dass für jedes $v \in \{0,1\}^{p(n)}$ zumindest ein Wort der Form $v \oplus u_i$ in $B(x)$ enthalten ist. Dann müssen die Mengen

$$B_i = \{v \mid v \oplus u_i \in B(x)\},$$

die alle gleichmächtig zu $B(x)$ sind, ganz $\{0,1\}^{p(n)}$ abdecken. Folglich muss $B(x)$ mindestens $2^{p(n)}/p(n)$ Wörter enthalten. Für hinreichend große n schließt dies jedoch die Zugehörigkeit von x zu $\overline{A}$ aus, da dann $B(x)$ höchstens $2^{-n}2^{p(n)}$ Wörter enthalten dürfte.

Die Zugehörigkeit von BPP zu Π_2^p ergibt sich unmittelbar aus dem Komplementabschluss von BPP. ■

Starten wir in obigem Beweis mit einer Sprache A in BP $\cdot$ co-NP (d.h. die Menge B gehört zu co-NP), dann folgt ebenfalls $A \in \Sigma_2^p$, da die Sprache

$$B' = \{x\#u_1\# \cdots \#u_{p(n)}\#v \mid \exists i : x\#(v \oplus u_i) \in B\}$$

in co-NP entscheidbar und $\exists \cdot \forall \cdot$ co-NP $= \Sigma_2^p$ ist.

Korollar 100
BP $\cdot$ co-NP $\subseteq \Sigma_2^p$ bzw. BP $\cdot$ NP $\subseteq \Pi_2^p$.

8 Das Graphisomorphieproblem

In diesem Kapitel wollen wir die Komplexität des Graphisomorphieproblems untersuchen.

Graphisomorphieproblem (GI):

Gegeben: Ungerichtete Graphen $G_i = (V, E_i)$, $i = 1, 2$ mit $V = \{1, \ldots, n\}$ und $E_i \subseteq \binom{V}{2}$.

Gefragt: Sind die beiden Graphen G_1 und G_2 isomorph?

Zur Erinnerung: Die beiden Graphen G_1 und G_2 sind isomorph (kurz: $G_1 \cong G_2$), falls eine Permutation $\varphi \in S_n$ mit

$$\{u, v\} \in E_1 \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E_2$$

für alle $u, v \in V$ existiert (ein solches φ heißt **Isomorphismus** zwischen G_1 und G_2; aus beweistechnischen Gründen setzen wir voraus, dass beide Graphen dieselbe Knotenmenge besitzen).

Proposition 101
GI $\in$ NP

Offene Fragen:

- Ist GI $\in$ P?
- Ist GI NP-vollständig?
- Ist GI $\in$ co-NP?

Eng verwandt mit GI ist das Problem, die Existenz eines nichttrivialen Automorphismus' in einem Graphen zu entscheiden.

Graphautomorphieproblem (GA):

Gegeben: Ein ungerichteter Graph $G = (V, E)$.

Gefragt: Besitzt G einen nichttrivialen Automorphismus?

Zur Erinnerung: Eine Permutation $\varphi \in S_n$ mit

$$\{u, v\} \in E \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E$$

für alle $u, v \in V$ heißt **Automorphismus** von G. Da jeder Graph zumindest einen Automorphismus besitzt (nämlich die Identität id), sind wir nur an so genannten nichttrivialen Automorphismen $\varphi \neq id$ interessiert. Mit den Bezeichnungen

$$\mathit{Aut}(G) := \{\varphi \in S_n \mid \varphi \text{ ist ein Automorphismus von } G\}$$

und

$$\mathit{Iso}(G_1, G_2) := \{\varphi \in S_n \mid \varphi \text{ ist ein Isomorphismus zwischen } G_1 \text{ und } G_2\}.$$

gelten demnach die Charakterisierungen

$$\mathrm{GI} = \{\langle G_1, G_2\rangle \mid \mathit{Iso}(G_1, G_2) \neq \emptyset\}$$

und

$$\mathrm{GA} = \{G \mid \mathit{Aut}(G) \neq \{id\}\}.$$

Lemma 102

Sei $G = (V, E)$ ein Graph mit $V = \{1, \ldots, n\}$. Für eine Permutation $\varphi \in S_n$ bezeichne $\varphi(G)$ den Graphen (V, E') mit

$$E' := \{\{\varphi(u), \varphi(v)\} \mid \{u, v\} \in E\}.$$

Dann gilt:

(i) $G = \varphi(G) \Leftrightarrow \varphi \in \mathit{Aut}(G)$,

(ii) $\{H \mid G \cong H\} = \{\varphi(G) \mid \varphi \in S_n\}$,

(iii) $\|\{H \mid G \cong H\}\| = \frac{n!}{\|\mathit{Aut}(G)\|}$.

Beweis:

1. Sei $\varphi(G) = (V, E')$. Dann gilt

$$\begin{aligned} G = \varphi(G) &\Leftrightarrow E = E' \\ &\Leftrightarrow \forall u, v \in V : \{u, v\} \in E \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E \\ &\Leftrightarrow \varphi \in \mathit{Aut}(G). \end{aligned}$$

2. Gelte $G \cong H$ und sei $H = (V, E')$. Dann existiert ein Isomorphismus $\varphi \in S_n$ zwischen G und H, d. h.

$$\forall u, v \in V : \{u, v\} \in E \Leftrightarrow \{\varphi(u), \varphi(v)\} \in \tilde{E}.$$

Also ist $H = \varphi(G)$. Die Inklusion von rechts nach links ist klar, da G und $\varphi(G)$ isomorph sind.

3. Wir nennen zwei Permutationen φ und π äquivalent, falls $\varphi(G) = \pi(G)$ ist. Da $\mathit{Aut}(G)$ eine Untergruppe von S_n ist, folgt

$$\begin{aligned} \varphi(G) = \pi(G) &\Leftrightarrow \varphi^{-1}(\pi(G)) = G \\ &\Leftrightarrow \varphi^{-1} \circ \pi \in \mathit{Aut}(G) \\ &\Leftrightarrow \varphi^{-1} \circ \pi \circ \mathit{Aut}(G) = \mathit{Aut}(G) \\ &\Leftrightarrow \varphi \circ \mathit{Aut}(G) = \pi \circ \mathit{Aut}(G). \end{aligned}$$

Zwei Permutationen sind also genau dann äquivalent, wenn sie in der gleichen Nebenklasse von $Aut(G)$ liegen, d.h.

$$\|\{H \mid G \cong H\}\| = \|\{\varphi \circ Aut(G) \mid \varphi \in S_n\}\|.$$

Aus der Gruppentheorie wissen wir jedoch, dass die Nebenklassen von $Aut(G)$ die Gruppe S_n in gleichmächtige Teilmengen partitionieren und daher genau $\frac{n!}{\|Aut(G)\|}$ verschiedene Nebenklassen existieren.

■

Als nächstes wollen wir zeigen, dass GI fast in co-NP liegt (genauer: GI $\in$ BP $\cdot$ co-NP bzw. $\overline{\text{GI}} \in$ BP $\cdot$ NP). Hierzu führen wir den modifizierten BP-Operator $\widetilde{\text{BP}}$ ein und zeigen die beiden folgenden Resultate:

- $\overline{\text{GI}} \in \widetilde{\text{BP}} \cdot \text{NP}$,
- $\widetilde{\text{BP}} \cdot \text{NP} = \text{BP} \cdot \text{NP}$.

Definition 103 ($\widetilde{\text{BP}} \cdot \mathcal{C}$)

Sei $\mathcal{C}$ eine Sprachklasse. Eine Sprache $L \subseteq \Sigma^*$ gehört zu $\widetilde{\text{BP}} \cdot \mathcal{C}$, falls Funktionen $g(x) \geq 1$ in FP und $f(x)$ in $\#\mathcal{C}$ existieren, so dass für alle $x \in \Sigma^*$ gilt:

$$\begin{array}{lcl} x \in L & \Rightarrow & f(x) \geq 2g(x) \\ x \notin L & \Rightarrow & f(x) \leq g(x). \end{array}$$

Es ist leicht zu sehen, dass NP in der Klasse $\widetilde{\text{BP}} \cdot$ P enthalten ist (hierzu genügt es, für g die konstante Funktion $g(x) = 1$ zu wählen und die Anzahl der akzeptierenden Pfade zu verdoppeln). Daher ist BPP vermutlich echt in $\widetilde{\text{BP}} \cdot$ P enthalten.

Satz 104

$\overline{\text{GI}} \in \widetilde{\text{BP}} \cdot \text{NP}$.

Beweis: Seien zwei Graphen $G_i = (V, E_i)$, $i = 1, 2$, mit $V = \{1, \ldots, n\}$ gegeben. Betrachte die Mengen

$$X(G_i) := \{\langle H, \pi\rangle \mid H \cong G_i \wedge \pi \in Aut(H)\},$$

wobei $|\langle H, \pi\rangle| = p(n)$ für ein festes Polynom p sei. Dann gilt

$$\|X(G_i)\| = \sum_{H, H \cong G_i} \underbrace{\|Aut(H)\|}_{=\|Aut(G_i)\|} = \frac{n!}{\|Aut(G_i)\|} \cdot \|Aut(G_i)\| = n!$$

und somit gilt für $X(G_1, G_2) := X(G_1) \cup X(G_2)$:

$$\begin{array}{lclcl} G_1 \cong G_2 & \Rightarrow & X(G_1) = X(G_2) & \Rightarrow & \|X(G_1, G_2)\| = n! \\ G_1 \not\cong G_2 & \Rightarrow & X(G_1) \cap X(G_2) = \emptyset & \Rightarrow & \|X(G_1, G_2)\| = 2n! \end{array}$$

Da die Sprache

$$B = \{\langle G_1, G_2\rangle \# \langle G, \pi\rangle \mid \langle G, \pi\rangle \in X(G_1, G_2)\}$$

in NP entscheidbar und die Funktion $g(G_1, G_2) = n!$ in FP berechenbar ist, folgt $\overline{\mathrm{GI}} \in \widetilde{\mathrm{BP}} \cdot \mathrm{NP}$. ■

Definition 105 $(Lin(n,k))$

$Lin(n,k)$ bezeichne die Menge aller linearen Funktionen von $\{0,1\}^n$ nach $\{0,1\}^k$.

Bemerkung 106

Jede Funktion $h \in Lin(n,k)$ lässt sich eineindeutig durch eine Matrix $A_h \in \{0,1\}^{k\times n}$ beschreiben, d.h. es gilt

$$h(y_1 \cdots y_n) = (z_1 \cdots z_k) \Leftrightarrow \underbrace{\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kn} \end{pmatrix}}_{A_h} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix}$$

$$\Leftrightarrow z_j := \underbrace{a_{j1}y_1 \oplus \cdots \oplus a_{jn}y_n}_{=:h_j(y)} \text{ für } j = 1, \ldots, k.$$

Lemma 107

Sei $\emptyset \neq B \subseteq \{0,1\}^n - \{0^n\}$ und für eine zufällig unter Gleichverteilung gewählte Funktion $h \in_R Lin(n,k)$ sei S die ZV

$$S = \left\| \left\{ y \in B \mid h(y) = 0^k \right\} \right\|.$$

Dann gilt

$$\begin{aligned} E(S) &= 2^{-k} \cdot \|B\|, \\ \mathit{Var}(S) &= 2^{-k} \cdot (1 - 2^{-k})\|B\|. \end{aligned}$$

Beweis: Betrachte für festes $y \in B$ und zufällig gewähltes $h \in_R Lin(n,k)$ die Zufallsvariable

$$S_y := \begin{cases} 1, & h(y) = 0^k \\ 0, & \text{sonst.} \end{cases}$$

Dann gilt

$$S = \sum_{y \in B} S_y.$$

Wegen $h(y) = h_1(y) \cdots h_k(y)$ gilt

$$\begin{aligned} \mathit{Pr}\left[h(y) = 0^k\right] &= \mathit{Pr}\left[h_j(y) = 0 \text{ für } j = 1, \ldots, k\right] \\ &= \prod_{j=1}^{k} \underbrace{\mathit{Pr}\left[h_j(y) = 0\right]}_{1/2} \\ &= 2^{-k}. \end{aligned}$$

$Pr[h_j(y) = 0] = 1/2$ gilt, da

$$\|\{h \in Lin(n,k) \mid h_j(y) = 0\}\| = \|\{h \in Lin(n,k) \mid h_j(y) = 1\}\|.$$

Dies lässt sich wie folgt erklären: Sei i ein Index mit $y_i \neq 0$. Dann wird durch Flippen des i-ten Bits von h_j (also von a_{ji} in A_h) eine Bijektion zwischen diesen beiden Mengen beschrieben. Also ist

$$E(S) = \sum_{y \in B} E(S_y) = 2^{-k}\|B\|$$

und wegen $S_y^2 = S_y$

$$Var(S_y) = E(S_y^2) - E(S_y)^2 = 2^{-k} - 2^{-2k} = 2^{-k}\left(1 - 2^{-k}\right) < E(S).$$

Es bleibt nur noch zu zeigen, dass die Zufallsvariablen S_x und S_y im Fall $x \neq y$ stochastisch unabhängig sind, und somit $Var(S) = \sum_{x \in B} Var(S_x)$ ist. Sei i eine Position mit $y_i = 1$ und $x_i = 0$ (falls nötig, vertauschen wir x und y). Dann ändert sich durch Flippen von a_{ji} in der Matrix A_h der Wert von $h_j(y)$, wogegen sich der von $h_j(x)$ nicht ändert. Folglich gilt für $j = 1, \ldots, k$,

$$\|\{h \mid h_j(x) = 0 \wedge h_j(y) = 0\}\| = \|\{h \mid h_j(x) = 0 \wedge h_j(y) = 1\}\|.$$

und daher

$$Pr[h_j(y) = 0 \mid h_j(x) = 0] = 1/2.$$

Dies impliziert wiederum

$$Pr[h_j(x) = h_j(y) = 0] = \underbrace{Pr[h_j(x) = 0]}_{1/2} \cdot \underbrace{Pr[h_j(y) = 0 \mid h_j(x) = 0]}_{1/2} = 1/4$$

und somit

$$\begin{aligned} Pr\left[h(x) = h(y) = 0^k\right] &= \prod_{j=1}^{k} Pr[h_j(x) = h_j(y) = 0] \;=\; 4^{-k} \\ &= Pr\left[h(x) = 0^k\right] \cdot Pr\left[h(y) = 0^k\right]. \end{aligned}$$

Also sind die Zufallsvariablen S_x und S_y stochastisch unabhängig. ■

Satz 108
BP · NP = $\widetilde{\text{BP}}$ · NP.

Beweis: Für die Inklusion von links nach rechts genügt es, für g die Funktion $g(x) = 2^{p(|x|)}/3$ zu wählen. Für die umgekehrte Inklusion sei A eine Sprache in $\widetilde{\text{BP}}$ · NP Dann existieren Funktionen $g(x) \geq 1$ in FP und $f(x)$ in #NP mit

$$\begin{aligned} x \in L &\quad\Rightarrow\quad f(x) \geq 2g(x), \\ x \notin L &\quad\Rightarrow\quad f(x) \leq g(x). \end{aligned}$$

Sei B eine NP-Sprache und sei p ein Polynom mit

$$f(x) = \|\{y \in \{0,1\}^{p(|x|)} \mid x\#y \in B\}\|,$$

wobei wir o.B.d.A. annehmen, dass B keine Wörter der Form $x\#0^m$ enthält. Für eine Eingabe x sei B'_x die Menge

$$B'_x = \{y_1 \dots y_5 \mid \text{ für } i = 1, \dots, 5 \text{ ist } |y_i| = p(|x|) \text{ und } x\#y_i \in B\},$$

d.h. $\|B'_x\| = f(x)^5$. Setzen wir $k(x) = \lceil \log_2(2^2 g(x)^5) \rceil$, so gilt

$$2^2 g(x)^5 \le 2^{k(x)} \le 2^3 g(x)^5 \text{ bzw. } {}^1\!/_8 \le 2^{-k(x)} g(x)^5 \le {}^1\!/_4.$$

Für eine zufällig aus $Lin(5p(|x|), k(x))$ gewählte Funktion h bezeichne S_x die Zufallsvariable

$$S_x = \|\{y \in B'_x \mid h(y) = 0^{k(x)}\}\|.$$

Dann ist $E(S_x) = 2^{-k(x)} \|B'_x\|$ und es gilt

$$\begin{aligned} x \in A &\Rightarrow \|B'_x\| \ge 2^5 g(x)^5 &&\Rightarrow E(S_x) \ge 2^{-k(x)} 2^5 g(x)^5 &&\Rightarrow E(S_x) \ge 4, \\ x \notin A &\Rightarrow \|B'_x\| \le g(x)^5 &&\Rightarrow E(S_x) \le 2^{-k(x)} g(x)^5 &&\Rightarrow E(S_x) \le {}^1\!/_4. \end{aligned}$$

Weiter folgt mit Tschebyscheff und $Var(S_x) = 2^{-k(x)}(1 - 2^{-k(x)})\|B'_x\| \le E(S_x)$

$$Pr[S_x = 0] \le Pr\Big[|S_x - E(S_x)| \ge E(S_x)\Big] \le \frac{Var(S_x)}{E(S_x)^2} \le \frac{1}{E(S_x)},$$

sowie

$$Pr[S_x \ge 1] = \sum_{i=1}^{\infty} Pr[S_x = i] \le \sum_{i=0}^{\infty} i \cdot Pr[S_x = i] = E(S_x).$$

Sei nun B'' die NP-Sprache

$$B'' = \{x\#h \mid h \in Lin(5p(|x|), k(x)) \text{ und } \exists y \in B'_x : h(y) = 0^{k(x)}\},$$

dann gilt für jede Eingabe x und für eine zufällig aus $Lin(5p(|x|), k(x))$ gewählte Funktion h,

$$\begin{aligned} x \in A \quad &\Rightarrow \quad E(S_x) \ge 4 \\ &\Rightarrow \quad Pr[x\#h \in B''] = Pr[S_x \ge 1] = 1 - \underbrace{Pr[S_x = 0]}_{\le 1/E(S_x)} \ge {}^3\!/_4, \\ x \notin A \quad &\Rightarrow \quad E(S_x) \le {}^1\!/_4 \\ &\Rightarrow \quad Pr[x\#h \in B''] = Pr[S_x \ge 1] \le E(S_x) \le {}^1\!/_4. \end{aligned}$$

Dies zeigt, dass A zu BP $\cdot$ NP gehört. ■

Lemma 109

Sei $\mathcal{C}$ abgeschlossen unter $\le_m$. Dann ist auch BP $\cdot$ $\mathcal{C}$ abgeschlossen unter $\le_m$.

Beweis: Sei B eine Sprache in BP $\cdot\,\mathcal{C}$ und gelte $A \leq_m B$ mittels einer Reduktionsfunktion $f \in$ FP. Wir müssen zeigen, dass dann auch A in BP $\cdot\,\mathcal{C}$ enthalten ist. Zu B existieren eine Sprache $B' \in \mathcal{C}$ und ein Polynom p mit

$$\|\{y \in \{0,1\}^{p(n)} \mid x \in B \Leftrightarrow x\#y \in B'\}\| \geq (2/3)2^{p(n)},$$

wobei n die Länge von x bezeichnet. Sei q ein Polynom mit $p(|f(x)|) \leq q(n)$ und sei A' die Sprache

$$A' = \{x\#y \mid |y| = q(n) \text{ und } f(x)\#y' \in B'\},$$

wobei y' das Präfix der Länge $p(|f(x)|)$ von y bezeichnet. Dann hat jedes Präfix y' mit der Eigenschaft

$$f(x) \in B \Leftrightarrow f(x)\#y' \in B',$$

wovon mindestens $(2/3)2^{p(|f(x)|)}$ existieren, genau $2^{q(n)-p(|f(x)|)}$ Verlängerungen y mit

$$x \in A \Leftrightarrow x\#y \in A'$$

und somit ist

$$\begin{aligned}\|\{y \in \{0,1\}^{q(n)} \mid x \in A \Leftrightarrow x\#y \in A'\}\| &\geq 2^{q(n)-p(|f(x)|)}(2/3)2^{p(|f(x)|)} \\ &= (2/3)2^{q(n)}.\end{aligned}$$

Also gehört A zu BP $\cdot\,\mathcal{C}$. ■

Korollar 110

Falls GI NP-vollständig ist, kollabiert PH auf die zweite Stufe,

$$\text{GI} \in \text{NPC} \Rightarrow \text{PH} = \text{BP} \cdot \text{NP} = \Sigma_2^P.$$

Beweis: Da mit NP auch BP$\cdot$co-NP unter $\leq_m$ abgeschlossen ist, folgt aus der Annahme GI $\in$ NPC, dass NP $\subseteq$ BP $\cdot$ co-NP ist. Wie wir gesehen haben impliziert dies wiederum den Kollaps PH $=$ BP $\cdot$ NP $=$ BP $\cdot$ co-NP. Wegen BP $\cdot$ co-NP $\subseteq \Sigma_2^P \subseteq$ PH folgt daraus PH $= \Sigma_2^P$. ■

9 Turing-Operatoren

Definition 111 (Orakel-Turingmaschinen)
Eine **deterministische Orakel-Turingmaschine (DOTM)** M ist eine DTM, die mit einem speziellen write-only **Orakelband** ausgerüstet ist. Außerdem besitzt M drei spezielle Zustände $q_?, q_+, q_-$. Als Orakel kann eine beliebige Sprache $A \subseteq \Sigma^*$ verwendet werden. Geht M in den **Fragezustand** $q_?$, so hängt der Folgezustand q' davon ab, ob das aktuell auf dem Orakelband stehende Wort y zu A gehört (in diesem Fall ist $q' = q_+$) oder nicht ($q' = q_-$). In beiden Fällen wird das Orakelband gelöscht und der Kopf an den Anfang zurückgesetzt. All dies passiert innerhalb eines einzigen Rechenschrittes. Die unter einem Orakel A arbeitende OTM wird mit M^A bezeichnet und die von M **unter Orakel** A **akzeptierte Sprache** mit $L(M^A)$.

Definition 112
Die **Rechenzeit** einer DOTM M bei Eingabe x ist

$$time_M(x) = \sup_{A \subseteq \Sigma^*} time_{M^A}(x).$$

M ist $t(n)$-**zeitbeschränkt**, falls für alle Orakel $A \subseteq \Sigma^*$ und alle Eingaben $x \in \Sigma^*$ gilt,

$$time_{M^A}(x) \leq t(|x|).$$

Eine polynomiell zeitbeschränkte DOTM M bezeichnen wir kurz als PDOTM. Alle unter einem Orakel A in Polynomialzeit akzeptierten Sprachen fassen wir in der Klasse

$$\mathrm{P}^A = \mathrm{P}(A) = \{L(M^A) \mid M \text{ ist eine PDOTM}\}$$

zusammen. P^A wird auch als die **Relativierung** der Klasse P zum Orakel A bezeichnet. Für eine Sprachklasse $\mathcal{C}$ sei

$$\mathrm{P}^{\mathcal{C}} = \mathrm{P}(\mathcal{C}) = \bigcup_{A \in \mathcal{C}} \mathrm{P}^A.$$

Genau so wie DTMs lassen sich auch NTMs und PTMs mit einem Orakelbefragungsmechanismus ausstatten, wodurch wir NOTMs und POTMs erhalten. Ist die Rechenzeit dieser Maschinen polynomiell beschränkt, so bezeichnen wir sie als NPOTMs bzw. PPOTMs. Entsprechend erhalten wir dann die relativierten Komplexitätsklassen NP^A, PP^A, BPP^A, RP^A, ZPP^A usw.

Satz 113
(i) $\mathrm{P}^{\mathrm{P}} = \mathrm{P}$ und $\mathrm{NP}^{\mathrm{P}} = \mathrm{NP}$,

(ii) $\mathrm{P}^{\mathrm{NP}\cap\text{co-NP}} = \mathrm{NP} \cap \text{co-NP}$ und $\mathrm{NP}^{\mathrm{NP}\cap\text{co-NP}} = \mathrm{NP}$,

(iii) $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$ und $\mathrm{NP}^{\Sigma_k^p} = \Sigma_{k+1}^p$ für $k \geq 0$.

Beweis: *(i)* Die Inklusion $\mathrm{P} \subseteq \mathrm{P}(\mathrm{P})$ ist klar. Für die umgekehrte Richtung sei L eine Sprache in P(P). Dann existiert eine PDOTM M und ein Orakel $A \in \mathrm{P}$ mit $L(M^A) = L$. Sei M' eine PDTM mit $L(M') = A$. Betrachte die TM M'', die bei Eingabe x die PDOTM M bei Eingabe x simuliert und jedesmal, wenn M eine Orakelfrage y stellt, die PDTM M' bei Eingabe y simuliert, um die Zugehörigkeit von y zu A zu entscheiden. Dann gilt

$$L(M'') = L(M^A) = L$$

und da $M(x)$ höchstens $time_M(x)$ Orakelfragen stellt,

$$\begin{aligned} time_{M''}(x) &= time_M(x) + time_M(x) \cdot \max_{y, |y| \leq time_M(x)} time_{M'}(y) \\ &= |x|^{\mathcal{O}(1)} \end{aligned}$$

Die Gleichheit von NP^{P} und NP lässt sich vollkommen analog zeigen.

(ii) Es reicht, $\mathrm{P}^{\mathrm{NP}\cap\text{co-NP}} \subseteq \mathrm{NP} \cap \text{co-NP}$ zu zeigen. Sei $L = L(M^A)$ für eine POTM M und sei A ein Orakel in $\mathrm{NP} \cap \text{co-NP}$. Dann existieren NPTMs M' und M'' mit $L(M') = A$ und $L(M'') = \overline{A}$. Betrachte die NPTM M^*, die bei Eingabe x die POTM M bei Eingabe x simuliert und jedesmal wenn M eine Orakelfrage y stellt, sich nichtdeterministisch dafür entscheidet, entweder M' oder M'' bei Eingabe y zu simulieren. In beiden Fällen wird die Simulation von M (im Zustand q_+ bzw. q_-) fortgeführt, wenn y von M' oder M'' akzeptiert wurde, anderfalls wird x verworfen. Dann gilt $L(M^*) = L(M^A) = L$ und daher ist $L \in \mathrm{NP}$. Da $\mathrm{P}^{\mathrm{NP}\cap\text{co-NP}}$ unter Komplementbildung abgeschlossen ist, ist dann $\mathrm{P}^{\mathrm{NP}\cap\text{co-NP}}$ auch in co-NP enthalten.

Die Inklusion von $\mathrm{NP}^{\mathrm{NP}\cap\text{co-NP}}$ in NP zeigt man vollkommen analog.

(iii) Wir zeigen zuerst die Inklusion von Σ_2^p in $\mathrm{NP}^{\mathrm{NP}}$. Zu jeder Sprache $L \in \Sigma_2^p = \exists \cdot \forall \cdot \mathrm{P}$ existiert eine Sprache $B \in \mathrm{P}$ und ein Polynom p mit

$$x \in L \Leftrightarrow \exists y \in \{0,1\}^{p(|x|)} \forall z \in \{0,1\}^{p(|x|)} : x\#y\#z \in B.$$

Definiere das NP-Orakel

$$A = \{x\#y \mid \exists z \in \{0,1\}^{p|x|)} : x\#y\#z \notin B\}$$

und betrachte die NPOTM M, die bei Eingabe x ein Wort $y \in \{0,1\}^{p(|x|)}$ rät, die Orakelfrage $x\#y$ stellt, und ihre Eingabe bei negativer Antwort akzeptiert und bei positiver Antwort verwirft. Offensichtlich akzeptiert M^A die Sprache L.

Für die umgekehrte Richtung sei M eine NPOTM, deren Rechenzeit durch ein Polynom p und deren Verzweigungsgrad durch 2 beschränkt ist. Weiter sei A ein NP-Orakel. Zu A existiert ein Polynom q und eine Sprache $B \in \mathrm{P}$ mit

$$y \in A \Leftrightarrow \exists z \in \{0,1\}^{q(|y|)} : y\#z \in B.$$

Nun können wir jede Rechnung von $M(x)$ durch ein Wort $r \in \{0,1\}^{p(|x|)}$ kodieren und es folgt

$$\begin{aligned} x \in L \quad \Leftrightarrow \quad & \exists r \in \{0,1\}^{p_M(|x|)} \exists b_1, \ldots, b_m \in \{0,1\} \\ & \exists y_q, \ldots, y_m \in \Sigma^{\leq p_M(|x|)} \\ & \exists z_1 \in \{0,1\}^{p(|y_1|)} \ldots \exists z_m \in \{0,1\}^{p(|y_m|)} \\ & \forall z_1' \in \{0,1\}^{p(|y_1|)} \ldots \exists \forall z_m' \in \{0,1\}^{p(|y_m|)} : \\ & R'(x, r, y_1, \ldots, y_m, z_1, \ldots, z_m, z_1', \ldots, z_m'), \end{aligned}$$

wobei R' die folgende Relation ist:

> r kodiert eine akzeptierende Rechnung von $M(x)$, während der die Orakelfragen $y_1, \ldots, y_m$ gestellt werden und mit $b_1, \ldots, b_m$ beantwortet werden ($b_i = 1$ bedeutet $y_i \in A$ und $b_i = 0$ bedeutet $y_i \notin A$), und für $i = 1, \ldots, m$ gilt:
>
> $$(b_i = 1 \wedge y_i \# z_i \in B) \vee (b_i = 0 \wedge y_i \# z_i \notin B).$$

Da die Gesamtlänge von $r, y_1, \ldots, z_m'$ polynomiell beschränkt ist in $|x|$, und da R' in Polynomialzeit entscheidbar ist, zeigt diese Charakterisierung von L, dass L in $\exists \cdot \forall \cdot$ P enthalten ist.

■

Satz 114

Es gibt Orakel A, B, so dass

$$\mathrm{P}^A = \mathrm{NP}^A \quad \wedge \quad \mathrm{P}^B \neq \mathrm{NP}^B.$$

Beweis: Wählen wir für A eine PSPACE-vollständige Sprache wie z.B. QBF, so gilt

$$\text{PSPACE} = \{L \mid L \leq \text{QBF}\} \subseteq \text{P(QBF)} \subseteq \text{NP(QBF)} \subseteq \text{NPSPACE} \subseteq \text{PSPACE}.$$

Es bleibt also nur noch eine Sprache $B \subseteq \{0,1\}^*$ mit $\mathrm{P}^B \neq \mathrm{NP}^B$ zu finden. Unabhängig davon wie wir B konstruieren wird die Testsprache

$$L(B) = \{0^n \mid \{0,1\}^n \cap B \neq \emptyset\}$$

sicherlich zu NP(B) gehören. Unser Ziel ist es, B mittels Diagonalisierung so zu konstruieren, dass $L(B)$ nicht in P^B enthalten ist.

Sei $M_1, M_2, \ldots$ eine Aufzählung von PODTMs, wobei wir annehmen können, dass die Laufzeit von M_i durch das Polynom $n^i + 1$ beschränkt ist,

$$time_{M_i}(x) \subseteq (|x|)^i + 1,$$

und für jede Sprache $L \in \mathrm{P}^B$ ein Index i existiert mit $L(M_i^B) = L$.

Wir konstruieren B stufenweise als Vereinigung von Sprachen B_i, wobei B_{i+1} aus B_i durch hinzufügen maximal eines Wortes y der Länge n_{i+1} entsteht und die Zahlenfolge n_i, $i \geq 0$, induktiv wie folgt definiert ist:

$$\begin{aligned} n_0 &= 0, \\ n_{i+1} &= \min\{n \geq (n_i)^i \mid n^{i+1} < 2^n\} + 1. \end{aligned}$$

Durch die Bedingung $(n_i)^i < 2^{n_i}$ stellen wir sicher, dass M_i bei Eingabe 0^{n_i} das Orakel nicht über alle Wörter der Länge n_i befragen kann. Nun können wir B_i in Stufe i so definieren, dass M_i nicht die Sprache $L(B)$ akzeptiert:

Stufe 0: $B_0 = \emptyset$.

Stufe $i, i \geq 1$: Falls $M_i^{B_{i-1}}$ die Eingabe 0^{n_i} akzeptiert, setze $B_i = B_{i-1}$. Verwirft dagegen $M_i^{B_{i-1}}$ diese Eingabe, setze $B_i = B_{i-1} \cup \{y\}$, wobei y das lexikografisch kleinste Wort der Länge n_i ist, das während dieser Rechnung nicht als Orakelfrage gestellt wird. Hierdurch erreichen wir, dass sich $M_i^{B_i}(0^{n_i})$ gleich wie $M_i^{B_{i-1}}(0^{n_i})$ verhält und $0^{n_i} \in L(M_i^{B_i}) \triangle L(B_i)$ ist.

Nun können wir uns leicht davon überzeugen, dass die Testsprache $L(B)$ für das Orakel $B = \bigcup_{i \geq 0} B_i$ nicht in P^B enthalten ist. Andernfalls müsste nämlich ein Index i existieren mit $L(M_i^B) = L(B)$. Da jedoch M_i^B bei Eingabe 0^{n_i} nur Orakelfragen y der Länge

$$|y| < \mathit{time}_{M_i}(0^{n_i}) \leq (n_i)^i + 1 \leq n_{i+1}$$

stellen kann, verhält sich M_i^B bei Eingabe 0^{n_i} wie $M_i^{B_i}(0^{n_i})$ und somit wie $M_i^{B_{i-1}}(0^{n_i})$, d.h. $0^{n_i} \in L(M_i^B) \triangle L(B)$. ■

Fast alle bisher in der Komplexitätstheorie erzielten Resultate wurden mit relativierbaren Beweistechniken erzielt und gelten daher relativ zu einem beliebigem Orakel. Beispiele hierfür sind alle in dieser Vorlesung gezeigten Inklusionen und Separierungen von Komplexitätsklassen wie

$$\mathrm{DTIME}^A(f) \subseteq \mathrm{NTIME}^A(f) \subseteq \mathrm{DSPACE}^A(f) \subseteq \mathrm{NSPACE}^A(f) \subseteq \mathrm{DTIME}^A(2^{O(f)}),$$

die Zeit- und Platzhierarchiesätze wie

$$\mathrm{DTIME}^A(g(n)) \subsetneq \mathrm{DTIME}^A(f(n)),$$

falls $g(n) \cdot \log g(n) = o(f(n))$, oder der Satz von Savitch,

$$\mathrm{NSPACE}^A(s(n)) \subseteq \mathrm{DSPACE}^A(s^2(n))$$

und der Satz von Immerman/Szelepczényi,

$$\mathrm{NSPACE}^A(s(n)) = \text{co-}\mathrm{NSPACE}^A(s(n)),$$

falls $s(n) \geq \log n$. Wie wir gerade gesehen haben, hängt dagegen die Antwort auf die Frage, ob $\mathrm{P}^A = \mathrm{NP}^A$ ist, von der Wahl des Orakels A ab. Daher müssen wohl erst neue nichtrelativierbare Beweistechniken entwickelt werden, um das $\mathrm{P} =?\mathrm{NP}$ Problem lösen zu können.